

Online Evaluation in WWW-based Courseware

Lúcio C. Tinoco, Edward A. Fox, N. Dwight Barnette
Virginia Polytechnic Institute & State University
Department of Computer Science
Blacksburg, VA 24061-0106
{tinoco | fox | barnette}@cs.vt.edu

Abstract

The QUIZIT system has been developed to support adaptive as well as standard testing, along with automatic grading and record keeping, using the WWW as a delivery vehicle. In connection with our *Interactive Learning with a Digital Library in Computer Science* project, we have applied QUIZIT to help with a master's level course as well as a large freshman level service course. Preliminary testing has led to various improvements, and it appears that QUIZIT will be well received by both instructors and students. It should port easily to a variety of environments and so could be widely used at other institutions as well.

Introduction

Since 1993, the *Interactive Learning with a Digital Library in Computer Science* project team has been revitalizing education in our department, offering a large repository of online CS courseware for outside use (at <http://ei.cs.vt.edu>). Millions of accesses per year take place, almost half from remote locations. With new server hardware now on order, as well as a large influx of general interest multimedia self-study modules recently added, we expect and encourage broader re-use of our WWW-available materials, that fill tens of thousands of files and relate to over 30 courses.

A key new contribution made in 1996 is our automated testing system, which will allow our students and those outside to evaluate their understanding of CS content, receiving instant feedback to aid learning, and yielding a lasting electronic record for instructors and project evaluation groups to interpret. This system also provides assistance with record keeping to faculty, which comes just in time as our department deals with the largest entering class ever, and increasing popularity of large service courses.

Motivation

Since interactive learning is our goal, the long turn around time in student evaluation and grading has been a continuing concern as we undertake distance learning and self-paced instruction [10]. Although electronic submission and grading of assignments by electronic mail (e-mail) speeds up the process considerably when compared to traditional paper submission, there is also a significant increase in the number of messages between instructors and students which can lead to queues and delays.

Automating part of the process of authoring and submitting exams is an efficient and low-cost alternative to achieve faster feedback in these cases. If used with moderation, and combined with other forms of evaluation, online quizzes can provide yet another means of assessing student performance while significantly reducing instructors' overload.

Existing Evaluation Systems

Most existing computerized test systems are part of computer based training (CBT) and multimedia authoring packages like Authorware [9], ToolBook [1], and many others. These systems normally come with graphical authoring interfaces, and course database support. Some of them also have networking capabilities, and provide extensive record keeping. Nevertheless, each of these systems has its own different "standards" (i.e., platforms, authoring languages, internal data representation). Further, they are usually very expensive to use on a large scale (e.g., in a university environment), and development costs also are very high. Because they all use different formats, test libraries cannot be easily integrated or accessed, reducing reuse and portability. Although some systems enable Internet deployment [1, 9] using HTML and Java, a content-based representation of the test documents is still missing.

With the popularity and growth of the World Wide Web (WWW) for delivering courseware [11], several CBT variations, often called web-based training (WBT), have begun to appear [8, 13]. Although inherently more portable and affordable than their counterparts, these systems have not yet achieved the same maturity. Most of them use HTML extensions as authoring languages, and focus more on presentational than on descriptive syntax. In

addition, database and record keeping support often is lacking and none of them provides the capability of authoring adaptable tests [3, 4].

In an effort to overcome the limitations of CBTs and WBTs mentioned above, we developed an automatic WWW-based adaptive quiz system called *QUIZIT* [12] to be used in connection with the courseware in the computer science department at Virginia Tech. The following sections go over *QUIZIT*'s main features, and explain how we overcame the problems of other existing systems.

The QUIZIT System Architecture

The *QUIZIT* system is composed of two distinct modules: the **authoring** module, which instructors can use to write their quizzes, and the **run-time** module. The run-time module is further sub-divided into two sub-modules: the **grader** module, (i.e., the students' interface), and the **monitor** module, used by instructors to keep track of student registration and performance. Figure 1 illustrates these modules and their interactions.

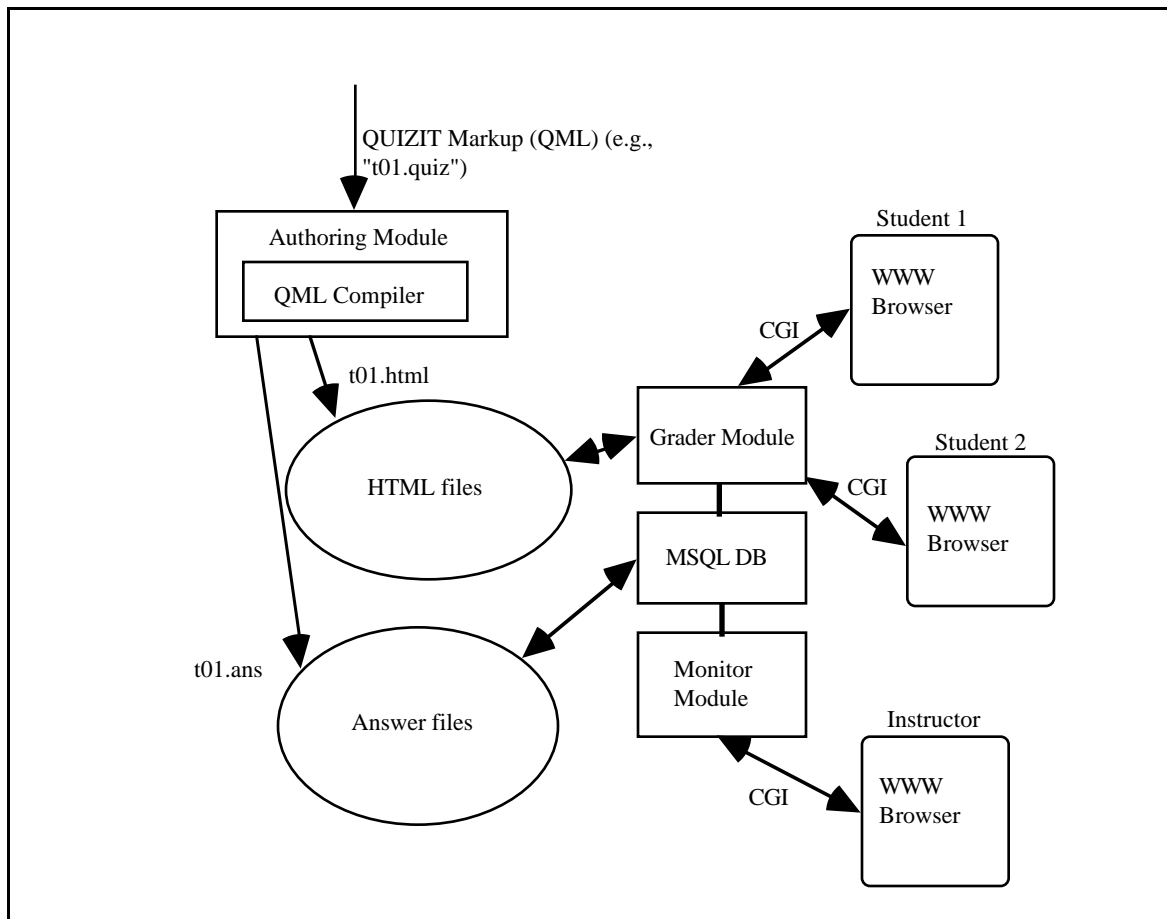


Figure 1. The QUIZIT System Architecture

The Authoring Module

Instructors write quizzes using a customized markup language, the QUIZIT Markup Language (QML), defined by a QUIZIT Document Type Definition (DTD). The use of the Standard Generalized Markup Language (SGML) [2] to define this DTD and the focus on a content-oriented, descriptive syntax (avoiding presentational elements) leads to a more usable and well-defined authoring process [12]. Quizzes also can have hyperlink capabilities defined in the language, as multiple levels of interaction may be desired (see *Adaptability* in the section below). HTML markup is allowed inside textual elements, like question descriptions

or general instructions. A special compiler, written in ANSI C using Lex and Yacc [7], translates QUIZIT markup into HTML pages and secure answer databases, which will be used later by the run-time module to instantly grade each quiz upon student submission.

Run-time Module: The Grader

The QUIZIT run-time engine is composed of several CGI programs (written in C) that interface the WWW pages generated by the compiler and the database engine used to record state information for each student. In addition, these

programs are responsible for grading and providing feedback to students after each quiz. Table 1 describes the type of feedback information found on the generated pages.

Feedback Info	What it means
score	Score on the quiz just submitted
cumulative score	Total score for all other quizzes including the one submitted
questions missed	A list of questions missed: highlighted, and with hyperlinks to a copy of the exam
time of submission	So that students can monitor the time spent on each quiz
next quiz to be taken	Depending on student's performance on the quiz just submitted.

Table 1. Information on the feedback page

Adaptability

One of the most important features of QUIZIT is to provide feedback in the form of **remedial** quizzes. By appropriately using "reply" elements in the markup language, instructors can define branching conditions for each quiz depending on the total score achieved. For example, if the quiz is worth 100 points, a instructor can define an easier remedial quiz for students who achieve less than, say, 50 points, and a harder quiz for students who pass that *cutoff* score. The run-time grader then takes care of fetching the right quiz after the student submits a quiz request using the browser.

Run-time Module: The Monitor

A WWW-based database interface also is provided for adding/deleting students, and for monitoring student performance during the evaluation process. The database server used is mSQL 1.0.16, a lightweight SQL server that runs under most UNIX systems [5]. The information stored in this database is summarized on Table 2.

At any given point in time, an instructor can browse the database information to monitor any student activity. A complete analysis of this information is extremely important during the assessment process, since many different paths and possibilities may arise because of the test's adaptability and/or other networking problems (i.e., time may or may not be so significant, since server connections can be very unstable). In particular, it is worth noticing that cumulative scores also may not be a good indicator of student's performance, and should be used carefully.

Database entry	What it means
ID	Student login ID (up to eight characters)
Password	Encrypted password for the student
Cookie	Session identifier (typically a social security number) for each student
Quiz	Current quiz
Next	Next quiz to be taken
Score	Score for the quiz being taken
Cumulative Score	Total score
Login time	Time when QUIZ was started
Submission time	Time the QUIZ was submitted

Table 2. Information in database

Preliminary Testing and PSI

Initial testing of QUIZIT has taken place in connection with CS5604, Information Storage and Retrieval, a course offered in various forms since the early 1970's. In 1993 this masters level course was revamped in accord with the Personalized System of Instruction (PSI) [6], but with administration aided by gopher and email; in 1994 courseware was shifted to WWW. Since record keeping and assessment require a great deal of instructors' time, as is often the case in PSI courses, a special test administration system was used in 1995. One of the aims of the QUIZIT effort was to help improve this situation even further in CS5604, as well as in several other PSI courses offered.

Our early testing has related to the first of 11 course modules, about digital libraries. A 104 page self-study tutorial on this "hot" topic was prepared to update materials used the previous year, along with a 6-part QUIZIT "test" (with 4-14 questions per part). These materials were used in connection with a tutorial on digital libraries given at ACM SIGIR'96 in Zurich during August 1996. Preparing the quiz took roughly the same time as preparing a regular quiz, though about 4 hours additional time was required to make this operational (as the first serious test of QUIZIT conversion and run-time software). To ensure good response time during the tutorial, which was given in a Sun lab at ETH, a Sun Voyager was connected to the local network to support interactive use of the WWW accessible tutorial notes and QUIZIT. Student were registered in the QUIZIT database as part of the operation of giving them individual passwords for the lab machines. They generally found QUIZIT easy to work with. Typical students spent 15-30 seconds per question. Given that tutorial attendees were mostly novices, and that the quiz questions were difficult, it is not surprising that scores were low (below 50%), but this was done deliberately to motivate further study of the extensive tutorial notes.

The same tutorial materials have been made available to students in CS5604. In addition, the 11 regular units of the course each have a quiz to ensure that students master unit content and satisfy unit objectives. Since students proceed at their own pace, some have completed the work sooner than others. In addition to monitoring the rate of progress, the instructor can determine who seems to have taken longer or made errors in the quizzes, to facilitate tutoring and other assistance. In general, after some initial startup problems (mostly related to passwords), use of QUIZIT has been relatively straightforward.

The new quizzes follow the schema shown in Figure 2. The aim is to divide each unit into two parts (a, b) along topical lines, and then to have three versions for each part. Since students will have a clear understanding of the topics covered in a part, if they have trouble with one of the versions they will be encouraged to study more before attempting the next version. This type of repeated study and evaluation should work even better than the system in place from 1995, which is illustrated in Figure 3. What students will lose in terms of moving from teaching assistant evaluation of short essays in 1995, they will gain from better organization of the questions into an adaptive framework with multiple parts and much faster feedback.

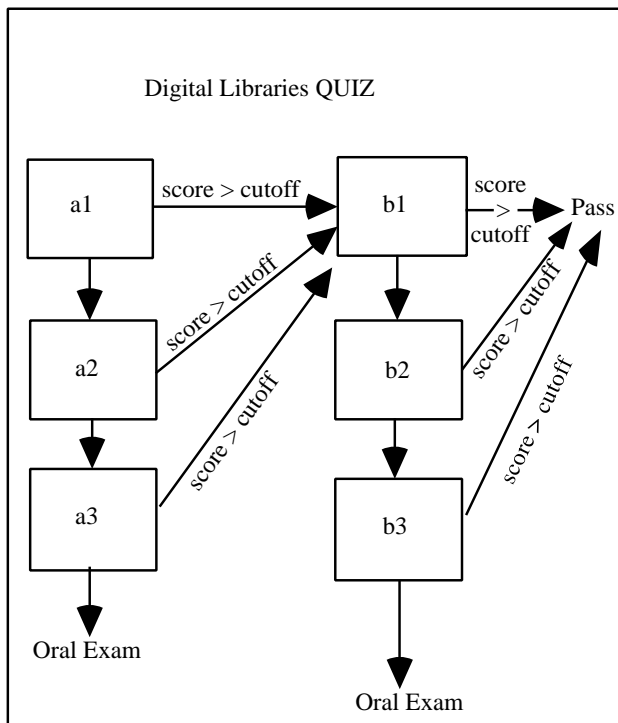


Figure 2: Structure of Digital Libraries (DL) Quiz – QUIZIT version.

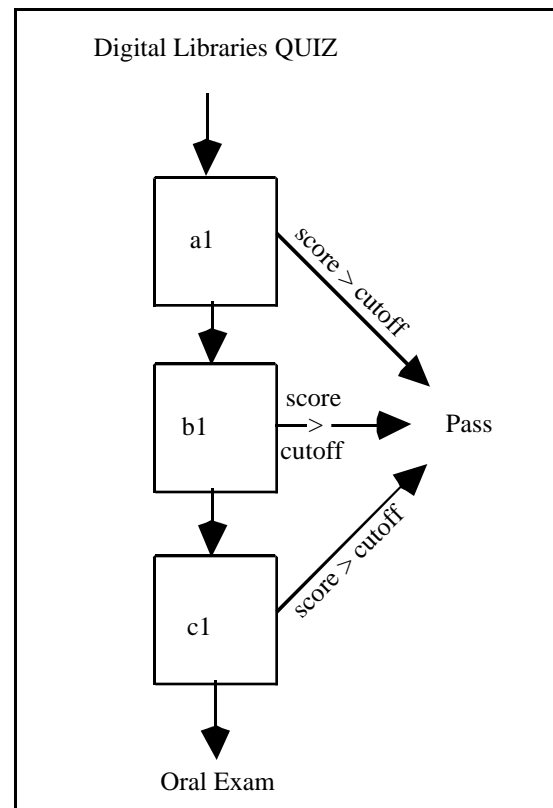


Figure 3: Structure of Digital Libraries (DL) Quiz – 1995 version.

The Case for QUIZIT in a Large Enrollment Class

QUIZIT is currently being utilized in CS1604 "Introduction to the Internet", at Virginia Tech. This is a large enrollment, one credit hour, introductory course, with a diverse student population. All course material (excluding the test) is available over the WWW at ["http://ei.cs.vt.edu/~netinfo/"](http://ei.cs.vt.edu/~netinfo/). Prior offerings of the course required students to attend lectures (presented with a WWW browser accessing online course material), submit assignments via email (graded and returned through email), and take in-class tests. Two sections of the course are currently being taught: 1. a traditional lecture version and 2. an independent study section. This will provide control and experimental groups to further evaluate the course, quizzes, tests, and the use of QUIZIT.

The Fall and Spring of '96 will be the last traditional lecture offerings of CS1604. Due to larger enrollment requests and an increased need for WWW familiarity by incoming students, automation of the course for independent study has become paramount. To this end QUIZIT has been adopted to attain these goals.

Traditional assignments and tests are being replaced by online quizzes and tests authored for and evaluated by QUIZIT. Course assignments are no longer mandatory or graded. Instead results from assignments will be

incorporated into online QUIZIT quizzes to evaluate assignment knowledge gains. In addition to the benefit of automatic evaluation, the most significant advantage is expected to be immediate feedback. Students will be able to instantaneously review their mistakes after a quiz or test, be informed of the correct answers, and access online course information to understand and rectify their incorrect information models.

The bi-weekly required quizzes allow students to proceed through the course material at their own pace. This will provide adaptability to different study speeds exhibited by various learner styles. It is expected that the monitoring of student progress, by both the instructor and student, will be greatly improved. Students will have instant access to their entire past course quizzes and tests for their own self evaluation purposes. Instructors shall be able to easily flag groups of marginally performing students for closer monitoring. An instructor also gains the ability to immediately ascertain the entire course population's progress and performance. Thus instructors have the singular opportunity to adjust the course during a term to tailor it to the current student body.

The authoring of quizzes and tests with QUIZIT featured powerful organization and presentation capabilities. Quizzes and tests were designed and arranged in a manner analogous to other test composition tools utilized in the past. The instructor found the ease of use of QUIZIT to be a pleasant surprise. A user is only required to have a familiarity with a text editor and a tagged language. An SGML editor allows for the macro expansions of the QUIZIT tags.

The remote, automated capabilities gained with QUIZIT in CS1604 has allowed for a very large scale (200+ students) Spring offering of the course with larger enrollments planned. Distance learning sections at other Virginia Tech campuses, Norfolk State University, Norfolk, Va. and other institutions, (Christopher Newport University, Newport News, Va. and Concord College, Athens, WV.), will also become a realization in the near future. The advantages derived from QUIZIT course integration initially appear to far outweigh any negative instructional side effects.

Conclusion and Future Work

In connection with a variety of efforts to improve the quality of CS education at Virginia Tech, the QUIZIT system has been developed to provide fast feedback to students as they take online quizzes, as part of traditional or Personalized System of Instruction courses. Developing quizzes with it takes about the same time as traditional quizzes, but then there are dramatic time savings when grading and evaluating student progress, as well as for the complicated record keeping required in proceed-at-your-own-pace courses. Thus we see improvements in both effectiveness and efficiency.

Future work will support more of the record keeping of PSI and other courses by adding new tables with the DBMS used with QUIZIT. Integrating this with the authoring and class administration tasks undertaken by instructors should provide further benefits. Tests in our usability labs will help us refine the QUIZIT interface, and see how well it can serve a diverse mix of CS students.

References

1. Asymetrix. *Toolbook CBT Edition*. [<http://www.asymetrix.com>]
2. Bryan, M., *SGML: An Author's Guide to the Standard Generalized Markup Language*. New York. Addison-Wesley, 1988.
3. ETS, *Registration Bulletin: GRE General Computer Based Test Descriptive Booklet*. Princeton, NY. Educational Testing Services Inc., 1995.
4. Hansen, D.N., ed. *An Investigation of Computer-Based Science Testing*. In *Computer-Assisted Instruction: A Book of Readings*, Ed. R.C. Atkinson and H.A. Wilson. Academic Press Inc. New York, NY, 1967.
5. Hughes Inc., *mSQL Manual*. [<http://Hughes.au.com/>], 1996.
6. Keller, F. S., Goodbye, teacher ... *J. of Applied Behavioral Analysis*, 1(1):79-89, Spring 1968.
7. Levine, J.R., T. Mason, and D. Brown, *Lex & Yacc*. 2nd edition. Sebastopol, CA. O' Reilly & Associates, Inc., 1992.
8. LRCorp. *LXRTest*. [<http://www.lxrtest.com/>], 1996.
9. Macromedia. *Authorware CBT*. [<http://www.macromedia.com>]
10. Mason, R., *Using Electronic Networking for Assessment*, in *Open and Distance Learning Today*, F. Lockwood, Editor. Routledge. London, 1995. 208-217.
11. McManus, T.F. *Delivering Instruction on the WWW*. [<http://www.edb.utexas.com/coe/depts/ci/it/projects/wbi/wbi.html>], 1995.
12. Tinoco, L., et al. QUIZIT: An Interactive Quiz System for WWW-based Instruction. *To appear in VII Symposium of Educational Technology, Belo Horizonte, Brazil*. Also in: [<http://pixel.cs.vt.edu/~ltinoco/quizit/QUIZIT.PS>]. (Post-script format), 1996.
13. Wheeler, D.A. *MkLesson User Guide*. [<http://lglwww.epfl.ch/Ada/Tutorials/Lovelace/userg.htm>] 1996.